The following learning outcomes will be assessed:

Knowledge:

1. Critical awareness of current problems and insights associated with modern web and database information systems development.

2. A comprehensive understanding of web and databases system development tools and technologies.

3. Critical understanding of data models, e.g. relational, NoSQL, and where they should be used.

Skills:

4. Competence in a variety of web and database development languages (e.g. Oracle, PHP) in order to develop interactive and professional data driven websites

Introduction

Chester's Cycling Outlet (CCO) Ltd. has grown from a small company based in a small market town, to one of the country's leading suppliers of bicycles and cycling equipment. However, they have never quite managed to move away from using a paper-based filing system for storing customer, sales, and stock information. To cope with their growth and allow for more efficiency, the company has decided to computerise their customer and stock management system. You have been tasked with developing a database application to meet their needs.

Current Position

Currently, CCO Ltd. record details of all customers (including their name, address, telephone number, date of birth and bank details, i.e., bank name, address, sort code and account number) who either purchase a bicycle or cycling equipment. Records are also kept of every purchase that takes place in any of the stores.

A stock warehouse is also kept. This makes it possible to see where any given product (including its identifier, type, name, description and cost) is currently stored. The stock warehouse also contains details of all purchases, and it allows stores to move stock from one store to another. This is particularly useful for those customers who wish to purchase a product which is not available in their local store.

The organisation would also like to be able to generate management reports, which may, for example, show all sales at a particular store or a group of stores, between certain dates.

PROPOSED SYSTEM

Using the Oracle Cloud ATP (Autonomous Transaction Processing) Relational Database Management System you are required to design and develop a prototype system that not only satisfies the requirements of the current system, but also has features that you consider to be worthwhile enhancements to the current system. To achieve this, you must:

a) Using an **Entity-Relationship (E-R) diagram**, produce a design of the proposed system, correctly showing labelled relationships with optionality and cardinality constraints clearly indicated. You must use the notation taught during the module. Ensure any many-to-many relationships are dealt with correctly in the diagram. Do not show attributes on the diagram, only entity names. Ensure that you clearly state any assumptions that you have made in creating your Entity-Relationship Diagram.

b) Create a set of tables and appropriate attributes, with sample data, for your design from part (a) above. Provide a **data dictionary** for each table which specifies:

i. table name.
ii. for each attribute, its name, description, data type and size where appropriate (using Oracle data types used in the SQL booklet for this module).
iii. primary key and any foreign keys (ensure you clearly specify which table the foreign key relates to).
iv. any other business constraints on the data (e.g., constraints on data values and dates, required format; and whether the attribute is null/not null).

c) Ensure that each table is fully normalised to 3NF. To do this you must **specify the functional dependencies and primary key for each table** and provide a statement as to why each table is in 3NF, clearly resolving any tables which are not in 3NF.

d) Subsequently, produce a **single SQL script file** which can be run from within Oracle Cloud without error and drops and:

i. drops and then creates your tables (correctly ensuring that any referential integrity issues can be resolved).
i. inserts sample data into every table (suggest 5-10 rows of data per table).
ii. creates the PL/SQL code described in part (e) below.
iii. includes test calls to your PL/SQL stored procedures supplying sample test data (ensuring you demonstrate a range of procedure calls including both valid and invalid test data examples). Do not prompt for user input, you just need to supply test data as parameters to your procedure calls.

e) Using Oracle PL/SQL develop PL/SQL stored procedures which allows:

i. an existing customer to purchase a single item from a specific store. This transaction must allow the client to specify an item (by its code) to be purchased, provide accurate cost information and deal with any problems, e.g., insufficient stock, etc. **Note that it is not necessary to create a shopping basket and there is no requirement to implement purchasing functionality such as linking to PayPal – you are simply required to store purchase details within the database.**

ii. the company to produce **a report showing all purchases** between two dates for a specific store. Make sure you show detailed information (by appropriately joining tables) in the resulting output, e.g., item name and description, not just an item id.

iii. the store **to move stock from one store to another**. You should supply a stock identifier and the store to move the stock from and to. The procedure must deal with any errors, e.g., if there is currently no stock of that item in the store.

Each PL/SQL stored procedure may require you to develop other PL/SQL stored functions, triggers and cursors that you think necessary to fully implement the required functionality.

Note:

You must not use any automated code (e.g., table definition) facilities of Oracle Cloud or any

other tool (e.g., SQL Developer) that you choose to use for any part of the assignment. Use

of any tools such as these will result in 0 marks for each affected element

When developing the system, you must consider the important development issues identified below:

General Development Issues:

The Management wishes their system to be extremely user-friendly. Therefore, you should consider

the following points:

• Appropriate use of messages, both error and informative, as well as user prompts should be

employed throughout.

• Automatic generation of record numbers (e.g., customer numbers, booking numbers, etc.) where

appropriate should be used.

• Ensure when considering data type sizes, that you consider realistic future growth of the organisation.

On a more technical level:

• Data types should match those used in the tutorial booklet (and those in your data dictionary).

• Dates and other relevant data should be validated accordingly.

• Exception Handling must be in place to deal with all errors, e.g., invalid dates, duplicate appointment times, insufficient stock, etc.

• Any fields that require mandatory input, i.e., NOT NULL must be validated on input

To be considered:

1-Excellent match to scenario, all labels, cardinality and optionality correctly specified. Assumptions clearly stated.

2-Very clearly laid out data dictionary. Excellent match to ERD. Table names match, very good specification of attributes and constraints.

3-Excellent specification of all FDs using appropriate notation. Clearly justified statement that matches 3NF. Any issues resolved correctly

4-Very good match to data dictionary. Would work completely in Oracle without error. Excellent sample data created. . PL/SQL code and PL/SQL procedure calls included.

5-Excellent code developed with excellent use of error checking and automated generation of data values. Very good set of test examples to demonstrate code running with both valid and invalid data.

6-Excellent code developed with excellent use of error checking and automated generation of data values. Very good set of test examples to demonstrate code running with both valid and invalid data.

7-Excellent code developed with excellent use of error checking and automated generation of data values. Very good set of test examples to demonstrate code running with both valid and invalid data.